Clustering Visuel Semi-Supervisé pour des systèmes en *coordonnées en étoiles* 3D

Loïc Lecerf*, Boris Chidlovskii*

*Xerox Research Centre Europe
6, chemin de Maupertuis,
38240 Meylan, France
{Prenom.Nom}@xrce.xerox.com,

Résumé. Dans cet article, nous proposons une approche qui combine les méthodes statistiques avancées et la flexibilité des approches interactives manuelles en clustering visuel. Nous présentons l'interface *Semi-Supervised Visual Clustering* (SSVC). Sa contribution principale est l'apprentissage d'une métrique de projection optimale pour la visualisation en *coordonnées en étoiles* ainsi que pour l'extension 3D que nous avons développée. La métrique de distance de projection est apprise à partir des retours de l'utilisateur soit en termes de similarité/dissimilarité entre les items, soit par l'annotation directe. L'interface SSVC permet, de plus, une utilisation hybride dans laquelle un ensemble de paramètres sont manuellement fixés par l'utilisateur tandis que les autres paramètres sont déterminés par un algorithme de distance optimale.

1 Introduction

Obtenir un clustering efficace et de haute qualité sur des données de grande taille est un problème majeur pour l'extraction des connaissances. Il existe une demande de plus en plus importante pour des techniques flexibles et efficaces de clustering capables de s'adapter à des jeux de données de structure complexe. Un ensemble de données est typiquement représenté dans un tableau composé de N items (lignes) et d dimensions (colonnes). Un item représente un événement ou une observation, alors qu'une dimension peut-être un attribut ou une caractéristique de l'item. Dans un mode *semi-supervisé* ou *supervisé*, une partie ou tous les items peuvent être annotés par une classe. Les méthodes de clustering tentent de partitionner les items en groupes avec une mesure de similarité. Un ensemble de données peut être grand en termes de nombre de dimensions, nombre d'éléments, ou les deux.

L'approche classique est basée sur des algorithmes de clustering, comme les K-moyennes, le clustering spectral ou hiérarchique ainsi que leurs multiples variantes (Hastie et al., 2001). Il existe cependant plusieurs inconvénients connus à ces méthodes. Premièrement, il n'est pas toujours facile de déterminer, visualiser et valider les clusters de forme irrégulière. Plusieurs algorithmes sont efficaces pour trouver des clusters dans des formes elliptiques (donc convenant aux distributions normales multidimensionnelles), mais peuvent échouer à reconnaître des clusters de forme complexe. Deuxièmement, les algorithmes existant sont automatiques, ils excluent toute intervention de l'utilisateur dans le processus jusqu'à la fin de l'algorithme.

Il n'y a aucune manière commode d'incorporer la connaissance du domaine au sein de la phase d'analyse ou de permettre à l'utilisateur d'orienter un processus de clustering lorsque l'on utilise des algorithmes automatisés. Typiquement, l'analyse de cluster continue après la fin de l'algorithme, jusqu'à ce que les utilisateurs soient satisfaits et acceptent le résultat. Cependant, lorsque le résultat n'est pas satisfaisant, les utilisateurs veulent être étroitement impliqués dans le processus itératif de clustering et d'évaluation, en fournissant leurs impressions et intuitions.

Une alternative aux méthodes automatiques de clustering est le *clustering visuel interactif* (Chen et Liu, 2004; Kandogan, 2001; Seo et Shneiderman, 2002). Ici les clusters sont visualisés sur un plan 2D ou un espace 3D ; cependant, la réduction de dimension nécessaire à la visualisation n'est pas effectuée en sélectionnant les dimensions "les plus importantes", mais par le principe du "*sweeping*" (Kandogan, 2001). Selon ce principe, les n dimensions sont représentées par n axes disposés sur le plan 2D ou l'espace 3D. Les coordonnées cartésiennes de chaque point de données étant alors définies en fonction de la direction et de la longueur de chaque axe. Un exemple récent de clustering visuel est le système iVIBRATE (Chen et Liu, 2006) basé sur le principe des *coordonnées en étoiles*. Son composant principal est le rendu visuel du clustering qui aide l'utilisateur dans le processus itératif de clustering au travers d'une visualisation interactive. Il permet de produire des solutions guidées par la visualisation pour traiter efficacement les clusters de formes irrégulières. Les résultats montrent que iVIBRATE peut réellement impliquer l'utilisateur dans le processus de clustering et générer des résultats de haute qualité sur de grands ensembles de données (Chen et Liu, 2006). À la différence des méthodes automatiques, le clustering itératif dans iVIBRATE est accompli essentiellement par la manipulation des paramètres (longueur et direction des axes). Cependant le clustering manuel devient difficilement possible lorsque que les données sont de grande dimension.

Dans ce papier, nous proposons de combiner les avantages des deux approches ci-dessus, l'analyse avancée des données des méthodes de clustering automatique et la flexibilité et l'interactivité du clustering visuel. Notre idée principale est l'apprentissage semi-supervisé d'une métrique de distance permettant une projection optimale pour les systèmes de visualisation en coordonnées en étoiles. De plus, nous avons étendu le principe des coordonnées en étoile 2D en 3D grâce à une disposition sphérique des axes. Cette extension améliore grandement le rendu visuel et facilite donc le clustering.

Dans la section suivante, nous présentons le principe des coordonnées en étoiles 2D et 3D. Puis, nous décrivons l'interface *Semi-Supervised Visual Clustering* qui enrichie le clustering visuel de l'apprentissage semi-supervisé de la métrique de distance optimale. Nous décrivons trois modes possibles disponibles dans l'interface : manuel, automatique et hybride. Les modes manuels et automatiques sont inspirés des méthodes automatiques et du clustering visuel interactif mentionnés précédemment. Nous présentons aussi une approche hybride dans laquelle certains paramètres sont manuellement fixés par l'utilisateur tandis que les paramètres restant sont déterminés par un algorithme de distance optimale, utilisant l'ensemble des retours de l'utilisateur. Nous concluons avec une évaluation de la métrique optimale sur la collection standard UCI.

2 Coordonnées en étoiles 2D et 3D

Les coordonnées en étoiles et leur extension sphérique placent respectivement les axes sur un plan 2D ou dans un espace 3D. Contrairement à la réduction de dimension, les axes de coor-

données ne sont pas nécessairement orthogonaux entre eux. La valeur minimale d'une donnée pour une dimension est tracée à l'origine, et la valeur maximale est tracée à l'extrémité de cet axe. Ainsi les vecteurs unités de chacun des axes sont calculés de manière à permettre la correspondance des valeurs des données à la longueur des axes. Les systèmes de visualisation de clustering visuel (Chen et Liu, 2006; Kandogan, 2001; Seo et Shneiderman, 2002) fournissent un certain nombre de dispositifs d'interaction, dont les utilisateurs peuvent se servir pour améliorer leur compréhension des données. Les fonctionnalités de bases du clustering visuel sont les suivantes :

Redimensionnement Le redimensionnement permet à des utilisateurs de changer la longueur d'un axe, en augmentant ou diminuant la contribution d'un attribut particulier sur la visualisation résultante.

Rotation La transformation de rotation modifie la direction du vecteur unité d'un axe, rendant un attribut particulier plus ou moins corrélé avec d'autres attributs (voir figure 1). Lorsque plusieurs axes sont tournés dans une même direction, leurs contributions sont agrégées dans la visualisation.

Annotation Les utilisateurs peuvent annoter des points soit par la sélection individuelle de points, soit par la sélection d'un ensemble de points au moyen d'une enveloppe convexe (voir figure 2). La couleur des points change selon l'annotation, ce qui permet de plus facilement les suivre dans la suite des transformations.

Point de vue En 3D, les utilisateurs peuvent changer de point de vue et zoomer sur des données pour trouver une meilleure visualisation des clusters.

Dans la suite, nous ferons principalement référence à l'extension 3D des coordonnées en étoiles.

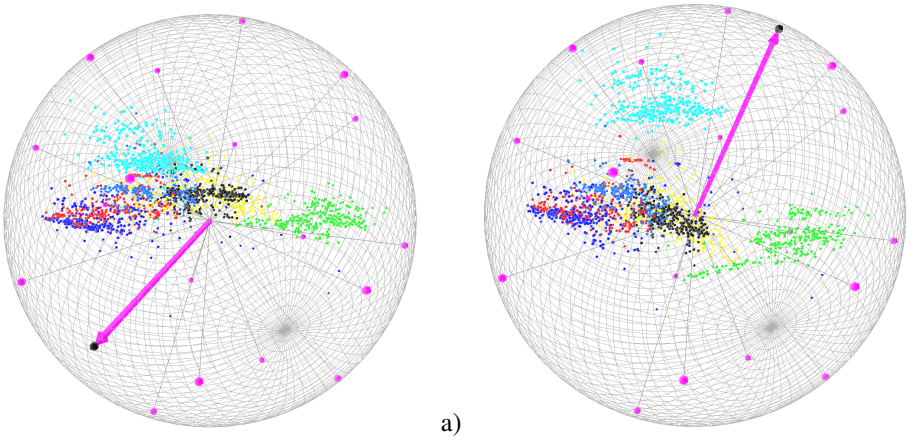

FIG. 1 – *Rotation d'un axe dans un SSVC : a) avant, b) après.*

2.1 Modélisation des Coordonnées en Étoile 3D (CE3D)

Le modèle de visualisation des Coordonnées en Étoile 3D (CE3D) se compose d'une normalisation $max - min$ suivi par une transformation A. La normalisation $max - min$ est

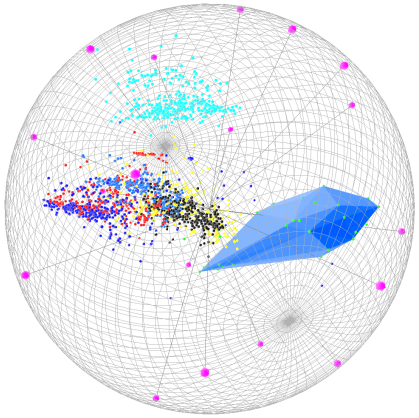

FIG. 2 – *Annotation d'un goupe dans SSVC par la sélection d'une zone convexe (bleu).*

utilisée pour normaliser les colonnes contenant des grandes valeurs. Pour une colonne ayant pour limite $[min, max]$, $max - min$ normalise une valeur v, dans une colonne comprise entre $[-1, 1]$ de la façon suivante : $v' = 2(v - min)/(max - min) - 1$ où v est la valeur originale et v' est la valeur normalisée. Ensuite, la transformation A place les points en d-dimension dans l'espace 3D selon la disposition des axes.

Soit un point 3D $Q(x, y, z)$ représentant l'image d'un point de donnée normalisée de dimension d, $P(x_1, \ldots, x_d)$ avec $x_i \in [-1, 1]$. $Q(x, y, z)$ est déterminé par la moyenne du vecteur somme des d vecteurs $sc_i \cdot x_i$, où sc_i sont les coordonnées sphériques qui représentent les d dimensions dans l'espace visuel 3D. Selon la transformation A, la projection 3D d'un point $Q(x, y, z)$ est déterminée par :

$$Q(x, y, z) = \frac{1}{d} \begin{pmatrix} \sum_{i=1}^{d} \alpha_i x_i cos(\theta_i) - x_0 \\ \sum_{i=1}^{d} \alpha_i x_i sin(\theta_i) sin(\phi_i) - y_0 \\ \sum_{i=1}^{d} \alpha_i x_i sin(\theta_i) cos(\phi_i) - z_0 \end{pmatrix} \tag{1}$$

Ici le vecteur $\alpha = [\alpha_1, \ldots, \alpha_d], \alpha_i \in [-1, 1]$, est issu des paramètres ajustables de *redimensionnement*, pour chacune des d dimensions. Ces α_i sont initialement fixés à 1. Les paramètres de *rotation* θ_i et ϕ_i sont initialement fixés à $2i\pi/d$ et peuvent être ajustés par la suite. Le point o=(x_0, y_0, z_0) fait référence au centre de l'espace de visualisation. La transformation A est une transformation linéaire avec un ensemble fixé de valeurs α, θ, ϕ. Si nous fixons le centre o la transformation $A_{\alpha,\theta,\phi}(x_1, \ldots, x_d)$ peut être représentée par la transformation $M\mathbf{x}^T$, dans laquelle :

$$M = \begin{bmatrix} \alpha_1 cos(\theta_1) & \cdots & \alpha_d cos(\theta_d) \\ \alpha_1 sin(\theta_1) sin(\phi_1) & \cdots & \alpha_d sin(\theta_d) sin(\phi_d) \\ \alpha_1 sin(\theta_1) cos(\phi_1) & \cdots & \alpha_d sin(\theta_d) cos(\phi_d) \end{bmatrix} \tag{2}$$

et $\mathbf{x} = [x_1, \ldots, x_d]$.

La transformation $A_{\alpha,\theta,\phi}(x_1, \ldots, x_d)$ est linéaire. Elle ne casse pas les clusters dans la visualisation. Ceci étant, l'écart visuel entre des nuages de points reflète un réel écart entre

les clusters dans l'espace original de grande dimension. Néanmoins, celle-ci peut-être la cause d'un chevauchement de clusters (Chen et Liu, 2006). La séparation de clusters imbriqués peut être accomplie grâce à la visualisation dynamique, au travers de la manipulation interactive. La transformation est ajustable par le redimensionnement des α_i et la rotation des θ_i et ϕ_i. Par la manipulation de ces paramètres, l'utilisateur peut voir l'influence de la $i$ème dimension sur la distribution des clusters au travers d'une série de changements de vue, lesquels sont d'importants indices pour le clustering. La figure 1 montre l'exemple du changement de forme d'un cluster par la rotation d'un axe (en gras).

Les dimensions importantes pour le clustering vont être la cause de changements importants dans la visualisation puisque les valeurs des paramètres correspondant sont changées de manière continue. Les axes sont disposés autour du centre d'affichage et les objets graphiques sont conçus pour ajuster interactivement chaque paramètre. Malheureusement, la conception visuelle limite tout de même le nombre de dimensions qui peuvent être visualisées et manipulées. La visualisation dans des systèmes en coordonnées en étoiles comme iVIBRATE, permettent aux utilisateurs de manipuler facilement jusqu'à une cinquantaine de dimensions.

3 Clustering Visuel Interactif Semi-Supervisé

Lorsque les données deviennent trop difficiles à manipuler, nous proposons d'améliorer le clustering visuel interactif par l'automatisation de certaines sous-tâches, si l'utilisateur pense que cela peut l'aider. Nous avons développé l'interface *Semi-Supervised Visual Clustering* (SSVC) laquelle implémente à la fois les coordonnées en étoiles et leur extension sphérique pour une visualisation sur un plan 2D ou un espace 3D. L'interface SSVC offre un processus de clustering interactif au travers de la manipulation de paramètres et un processus d'optimisation basé sur le rendu visuel 2D ou 3D. Le système itère jusqu'à ce que les utilisateurs soient satisfaits. Les principaux composants et réglages du processus sont discutés ci-dessous en détails.

Sélection de dimensions. Si l'ensemble de données contient trop de dimensions, les utilisateurs peuvent vouloir préliminairement écarter les moins pertinentes. Le nombre de dimensions peut en effet influencer la facilité de manipulation des paramètres et réduire la complexité des algorithmes. Ci-dessous nous considérons trois options pour la sélection d'attributs :

Manuel. L'utilisateur peut sélectionner ou désélectionner manuellement une ou plusieurs dimensions.

Mode non supervisé. L'analyse en composantes principales (ACP) (Hastie et al., 2001) peut être utilisée pour réduire la dimension des données par une approche linéaire. L'ACP ordonne les dimensions selon les valeurs propres de la matrice de covariance et permet de sélectionner une par une les $d' < d$ meilleures dimensions.

Mode semi-supervisé. Si une partie des données est déjà annotée, l'analyse discriminante de Fisher peut être utilisée pour réduire les dimensions par une approche linéaire. L'interaction basée sur l'entropie entre les attributs et les classes d'annotation (Yu et Liu, 2004) peut être une alternative non-linéaire, pour ordonner les dimensions et sélectionner les $d' < d$ meilleurs.

Manipulation et optimisation. Le processus de clustering interactif couvre essentiellement trois actions différentes. *Le mode manuel* est analogue aux systèmes de visualisation existant,

lorsque l'utilisateur ajuste manuellement les valeurs α, θ et ϕ pour modifier les clusters visibles. *Le mode automatique* fait référence au cas où les paramètres α, θ, ϕ sont déterminés selon la métrique de distance optimale apprise dans le mode semi-supervisé à partir des annotations disponibles et des retours de l'utilisateur. *Le mode hybride* fait référence à divers cas intermédiaires lorsque l'utilisateur fixe quelques α_i, θ_i, ϕ_i et exige que les paramètres restant soient déterminés par une métrique de distance optimale.

4 Clustering semi-supervisé

Le clustering semi-supervisé suppose qu'une petite quantité de données annotées est disponible pour un meilleur clustering. Ces données souvent proviennent des retours de l'utilisateur, soit sous la forme d'annotations directes d'un item par une classe, soit d'indication "plus légères" sur la similitude ou la dissimilitude de paires d'éléments. En utilisant ces indices, un meilleur clustering peut être réalisé par l'ajustement de la métrique de distance (Basu et al., 2004; Tang et al., 2007; Xing et al., 2003). Cette ajustement cherche à obtenir la vue utilisateur dans laquelle les items sont mis ensemble ou séparément. La représentation des données d'origine ne peut pas être incluse dans un espace où les clusters ne sont pas suffisamment séparés. Modifier la métrique de distance transforme cette représentation de sorte que les distances entre des éléments de même cluster sont réduites au minimum, alors que les distances entre des éléments de clusters différents sont maximisées.

Cependant, à la différence de Basu et al. (2004) qui apprend la métrique de distance dans l'espace d-dimensionnel original, nous visons ici l'espace visualisé en CE3D. L'utilisateur peut alors exprimer au mieux ses impressions et intuitions car ce n'est pas l'espace d'origine qui est optimisé mais la vue qu'il en a. Nous unifions donc ainsi dans un même système le clustering semi-supervisé et le clustering visuel 3D. En d'autres termes, nous recherchons une métrique de distance de projection optimale donnée par la matrice M en (2). Dans la matrice, la projection 3D du point **x** avec la projection M est donnée par $Q(x, y, z) = M\mathbf{x}$. Ci-dessous, nous considérons plusieurs alternatives pour la modélisation et l'évaluation de la métrique de distance de projection optimale.

4.1 Coordonnées sphérique vs analyse factorielle discriminante

Si un sous ensemble d'éléments est annoté, les coordonnées sphériques des axes pour la métrique de distance de projection optimale M peuvent-être obtenues par l'utilisation de l'analyse factorielle discriminante (Bishop, 1995) lesquelles sont une généralisation de l'analyse discriminante de Fisher à plus de 2 classes et 1 dimension projetée. Selon Bishop (1995), nous obtenons les variables discriminantes pour la projection 3D comme suit :

- Pour chaque classe nous formons la matrice de covariance V_k et sa moyenne μ_k. Puis nous définissons les poids de la matrice de covariance $V = \sum_{k=1}^{c} N_k V_k$, où N_k est le nombre d'éléments dans la classe k, et c est le nombre totale de classes.
- En utilisant la moyenne μ de l'ensemble des données et μ_k, la moyenne pour chacune des classes k, nous formons la matrice $V_B = \sum_{k=1}^{c} N_k(\mu_k - \mu)(\mu_k - \mu)^T$.
- Pour projeter dans l'espace 3D, nous construisons la matrice de projection optimale W_3 avec les 3 premiers vecteurs propres de $V^{-1}V_B$. Notons qu'il peut être coûteux sur un ensemble de grande dimension d'obtenir des vecteurs propres pour $V^{-1}V_B$.

Une fois la matrice de projection $W_3 = \{w_{ij}\}, i = 1, 2, 3, j = 1, \ldots, d$ obtenue, nous résolvons la matrice équation $M^T = W_3$ par la décomposition :

$$\left\{ \begin{array}{rcl} \alpha_i \cdot cos(\theta_i) & = & w_{i1}, \\ \alpha_i \cdot sin(\theta_i) sin(\phi_i) & = & w_{i2}, \\ \alpha_i \cdot sin(\theta_i) cos(\phi_i) & = & w_{i3}, i = 1, \ldots, d. \end{array} \right. \tag{3}$$

Si w_{i1}, w_{i2}, w_{i3} ne sont pas tous égaux à zéro, alors il existe une solution unique pour α_i, θ_i et ϕ_i dans (3). La solution est analogue pour la conversion des coordonnées cartésiennes aux coordonnées sphériques : $\alpha_i = \sqrt{w_{i1}^2 + w_{i1}^2 + w_{i3}^2}$, $\theta_i = arctan \frac{w_{i2}^2 + w_{i3}^2}{w_{i3}}$, $\phi_i = arctan \frac{w_{i2}}{w_{i3}}$. En d'autres termes, si la matrice de projection W_3 n'a pas de colonne 0, alors il existe un ensemble unique de θ, α et ϕ pour la visualisation en CE3D.

4.2 Apprendre la métrique de distance de projection

L'analyse discriminante de Fisher à partir des matrices de covariance, présentées précédemment, fait l'hypothèse implicite que la distribution des données est multinomiale. Dans cette section, nous suivons (Xing et al., 2003; Basu et al., 2004) en ne faisant aucune hypothèse spécifique sur la distribution des données. La métrique de distance est obtenue à partir de l'ensemble des paires de similitude/dissimilitude par l'optimisation d'une fonction qui cherche à réduire les distances des éléments semblables tout en augmentant les distances entre éléments différents. Cependant, à la différence de (Xing et al., 2003; Basu et al., 2004), nous recherchons une métrique de projection de distance M qui sépare les données dans la projection 3D plutôt que dans l'espace initial. Nous définissons la métrique de distance de projection M comme la distance entre deux éléments $\mathbf{x}_1$ et $\mathbf{x}_2$ dans l'espace 3D projeté :

$$d_M(\mathbf{x}_1, \mathbf{x}_2) = ||\mathbf{x}_1 - \mathbf{x}_2||_M = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T M^T M (\mathbf{x}_1 - \mathbf{x}_2)}. \tag{4}$$

Nous supposons que nous disposons d'un ensemble S de pairs de similarité ou de dissimilarité D. Si nous possédons les classes d'annotation pour quelques items, alors les items ayant la même classe forme l'ensemble de similarité S et les items ayant des classes différentes forme l'ensemble de dissimilarité D.

4.3 Problème d'optimisation pour la métrique de distance de projection

Trouver la métrique de distance de projection peut être formulé comme un problème d'optimisation non contrainte. Nous avons deux termes sur les ensembles S et D. Ces deux termes sont le terme *intra-groupe* In(M) pour toutes les paires de similitude données par l'ensemble S et le terme *extra-groupe* Ex(M) pour toutes les paires de dissimilarité données par D :

$$\begin{array}{rcl} \text{In}(M) & = & \sum_{(x_i, x_j) \in S} ||\mathbf{x}_1 - \mathbf{x}_2||_M^2, \\ \text{Ex}(M) & = & \sum_{(x_i, x_j) \in D} ||\mathbf{x}_1 - \mathbf{x}_2||_M^2. \end{array} \tag{5}$$

Nous considérons ensuite une fonction d'optimisation $J(M)$ qui cherche à diminuer In(M) et augmenter Ex(M). Nous prenons en considération le fait que les ensembles S et D peuvent être de tailles différentes, en particulier dans le cas de l'annotation multi-classes. Ci-dessous nous considérons 4 variantes de la fonction d'optimisation $J(M)$:

1. $J_1(M) = \text{In}(M) - \frac{|S|}{|D|}\text{Ex}(M)$,
2. $J_2(M) = \text{In}(M) - \frac{|S|}{|D|}log\text{Ex}(M)$,
3. $J_3(M) = \text{In}(M) + \frac{|S|}{|D|\text{Ex}(M)}$,
4. $J_4(M) = \frac{\text{In}(M)}{\text{Ex}(M)}$.

Dérivées partielles de $J(M)$. In(M) et Ex(M) ont tous deux des dérivées partielles. Pour développer les dérivées partielles pour le terme intra-groupe In(M), nous représentons la matrice M comme $M = \{m_{kl}\}, k = 1, 2, 3, l = 1, \ldots, d$. Puis nous réécrivons In(M) comme ci-dessous :

$$\text{In}(M) = \sum_{n,l=1}^{d} \sum_{k=1}^{3} \beta_{ln}^{S} m_{kl} m_{kn}, \tag{6}$$

Où $\beta_{ln}^{S} = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} (x_l^i - x_l^j)(x_n^i - x_n^j)$. Pour obtenir les dérivées, nous utilisons la symétrie des valeurs de β_{lk} $\beta_{ln}^{S} = \beta_{nl}^{S}, n, l = 1, \ldots, d$. Nous avons :

$$\frac{\partial \text{In}}{\partial m_{kl}} = 2 \sum_{n=1}^{d} \beta_{kn} m_{kn}, k = 1, 2, 3, l = 1, \ldots, d. \tag{7}$$

Les dérivées partielles pour Ex(M) dans toutes les variantes de $J(M)$ sont développées de manière similaire. Nous pouvons ensuite utiliser la méthode de descente du gradient ou bien une de ses variantes pour trouver la valeur du minimum (local) $J(M)$ (Basu et al., 2004).

4.4 Optimisation basée sur les contraintes

L'optimisation non contrainte présentée dans la sous-section précédente semble être sensible au déséquilibre de taille des ensembles S et D. Pour remédier à ce problème, nous allons adapter l'optimisation basée sur les contraintes proposée dans Basu et al. (2004) au CE3D. Dans cette approche, la fonction $J(M)$ tente de minimiser le terme intra-groupe In(M) tout en gardant le terme extra-groupe supérieur à 1 afin d'éviter la solution triviale $M = 0$ lorsque toutes les variables tombent à 0.

$$\begin{array}{ll} \min_A & \sum_{(x_i,x_j)\in S} ||\mathbf{x}_i - \mathbf{x}_j||_M^2, \\ avec & \sum_{(x_i,x_j)\in D} ||\mathbf{x}_i - \mathbf{x}_j||_M^2 \geq 1. \end{array} \tag{8}$$

Le choix de la constante 1 est arbitraire mais n'a pas de réelle importance. La changer pour n'importe qu'elle constante m conduit à remplacer M par $\sqrt{m}M$. En final, nous considérons une solution alternative, spécifique aux CE3D. Dans cette alternative, la fonction d'optimisation reste la même que dans (5). Néanmoins afin d'eviter les chevauchements de clusters, nous ajoutons des contraintes pour exclure les distributions de θ_i qui tendent à ramener tous les axes vers une direction unique, d'où :

$$\begin{array}{ll} \min_M & \sum_{(x_i,x_j)\in S} ||\mathbf{x}_i - \mathbf{x}_j||_M^2 - \frac{|S|}{|D|} \sum_{(x_i,x_j)\in D} ||\mathbf{x}_i - \mathbf{x}_j||_M^2 \\ avec & ave(|sin(|\theta_i - \theta_j|)|) > th, \end{array} \tag{9}$$

Où th est un seuil, $0 < th < 1$.

4.5 Apprentissage partiel de la métrique de distance

Les problèmes d'optimisation discutés dans les sections précédentes font référence au mode entièrement automatique du clustering visuel interactif. Le dernier mode, *hybride* fait référence aux cas où les paramètres sont partiellement définis, c'est à dire lorsque quelques (α_i, θ_i,ϕ_i) sont fixés par l'utilisateur, et que l'optimisation de la métrique de distance de projection M est limitée aux axes libres. La mise en place de l'apprentissage partiel est directe. Dans toutes les méthodes d'optimisation, tous les (α_i, θ_i, ϕ_i) sélectionnés sont fixés et l'optimisation est alors réalisée sur l'ensemble des variables laissées libres.

La figure 3 montre un exemple de clustering visuel pour l'ensemble de données `segment` issu de la collection UCI. La figure 3.a montre le rendu visuel initial en CE3D. La figure 3.b est le résultat d'une réduction de dimensions réalisée grâce à l'analyse discriminante de 21 à 4 dimensions, suivie par l'application de la métrique de distance obtenue par l'optimisation de (9).

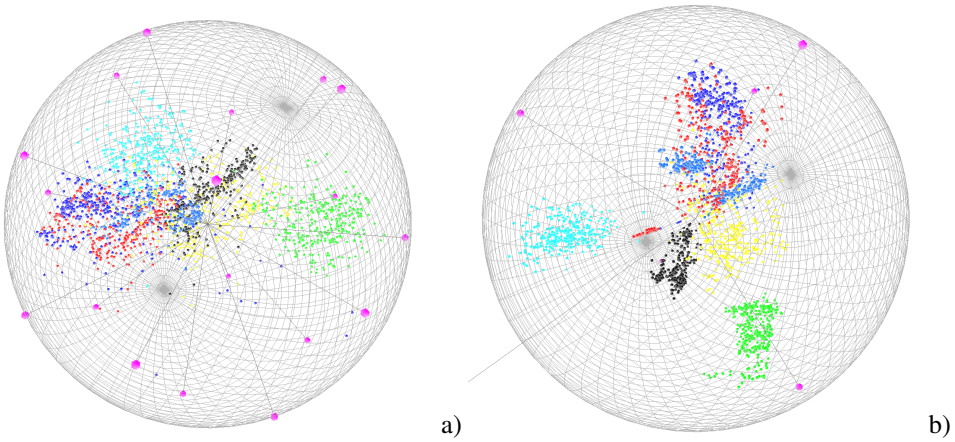

FIG. 3 – *Rendu visuel des données* `segment` *dans SSVC : a) état initial, b) état final.*

5 Intégration et évaluation

L'application initiale et principale que nous avons réalisée pour l'interface SSVC est l'assistance dans la classification de documents et la tâche d'annotation. La figure 3 montre le composant de SSVC intégré avec le système ALDAI (*Active Learning Document Annotation Interface*) d'annotation sémantique de document semi-structuré par apprentissage actif (Chidlovskii et al., 2006). Les éléments du document (les lignes dans notre cas) sont projetés dans l'espace 3D en utilisant une des métriques présentées précédemment. Chaque élément du document est associé avec le point correspondant dans l'interface SSVC ; les deux flèches noires dans la figure 4 indiquent un exemple d'association (élément du document, rendu dans SC).

Au delà de l'intégration, nous avons exécuté une série de tests pour l'évaluation des techniques d'optimisation présentées dans les sections précédentes. Dans l'interface SSVC, toutes les méthodes basées sur l'optimisation ont été implémentées en utilisant la librairie python

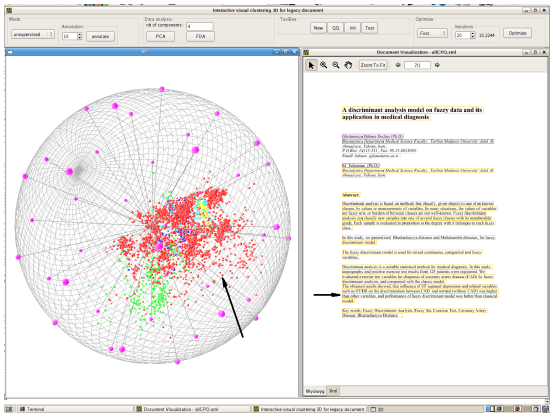

FIG. 4 – *Liens entre la visualisation en CE3D et l'annotation de document.*

	J_2	J_1	J_3	J_4
anneal	0.55322	0.55500	0.55322	**0.52032**
autos	0.43666	**0.41992**	0.43793	0.55599
breast	0.05421	**0.01507**	0.05411	0.01513
credit-a	0.13496	0.14244	**0.13390**	0.13406
glass	0.40067	**0.22544**	0.38041	0.42730
heart-c	0.18737	**0.15321**	0.18804	0.16892
hepatitis	0.15298	**0.13026**	0.15298	0.13937
iris	**0.00675**	0.01327	0.00677	0.00900
labor	0.27014	0.05058	**0.03468**	0.05996
lymph	0.42535	0.36257	0.42535	**0.25323**
segment	**0.08877**	0.29757	0.08877	0.33905
splice	0.33758	**0.32800**	0.33245	0.39398
Average	*0.25405*	**0.22444**	*0.23238*	*0.25136*

TAB. 1 – *Métrique de distance par optimisation non contrainte sur la collection UCI.*

pour l'optimisation numérique `scipy.optimize`. Nous avons testé les algorithmes sur une collection de 12 ensembles de données disponibles sur *UCI university ML repository* (`http ://www.ics.uci.edu`). Chaque méthode présentée dans la section 4 est évaluée en utilisant la fonction d'erreurs pondérées. Pour chaque méthode testée, nous obtenons la matrice M et nous déterminons les coordonnées pour tous les éléments. Nous appliquons alors l'algorithme de clustering choisi. Nous évaluons les résultats du clustering en considérant toutes les données placées dans un mauvais cluster. Chaque donnée est pondérée par la différence des distances entre le centroïde du mauvais et du bon cluster et nous normalisons par la distance entre ces deux centroïdes.

Ci-dessous nous présentons les résultats de l'évaluation pour tous les ensembles de données de la collection UCI avec toutes les techniques présentées dans les sections précédentes. Le tableau 1 reporte les résultats d'évaluation pour l'optimisation non contrainte et le tableau 2 reporte les résultats pour l'optimisation contrainte avec différentes valeurs de seuil th et la projection par analyse discriminante. Dans ce dernier cas, la projection est précédée d'une

	$J2, th = 0.1$	$J1, th = 0.1$	$J1, th = 0.2$	$J1, th = 0.3$	PCA 3 + FDA	PCA min(d,8) + FDA
anneal	0.4359	0.4504	0.4453	**0.3904**	0.39791	0.4721
autos	0.5053	0.4492	**0.3722**	0.3994	0.3510	0.3605
breast	0.0241	0.0161	0.0146	0.0143	**0.0109**	0.0117
credit-a	0.1317	0.1173	0.1119	**0.0895**	0.1194	0.1112
lass	0.3769	**0.1877**	0.1883	0.2577	0.3186	0.2266
heart-c	0.1796	0.1382	0.1209	0.1716	0.1365	**0.0843**
hepatitis	0.2103	0.1403	0.1497	**0.1083**	0.1403	0.1104
iris	0.0068	**0.0026**	0.0035	0.0075	0.0164	0.0184
labor	0.1935	0.0560	0.0668	0.0639	0.0439	**0.0040**
lymph	0.2693	**0.1147**	0.1537	0.1724	0.1232	0.1576
segment	**0.0884**	0.2477	0.2406	0.2768	0.4052	0.3406
splice	0.3376	0.3954	0.3377	0.3188	**0.0979**	0.2251
Average	*0.2299*	*0.1929*	*0.1838*	*0.1893*	*0.1748*	**0.1662**

TAB. 2 – *Métrique de distance par optimisation contrainte et analyse discriminante sur la collection UCI.*

sélection de 3 et $min(d, 8)$ attributs durant une étape préliminaire.

Il n'est pas surprenant de voir qu'en moyenne, les méthodes basées sur l'analyse discriminante se comportent mieux. Pour expliquer ce phénomène, nous avons étudié de plus près la collection elle-même ainsi que la fonction d'évaluation. En effet toutes les deux sont plus adaptées aux méthodes de clustering automatique : les groupes d'éléments dans les données ont une forme elliptique et la fonction d'évaluation favorise les résultats de clustering basés sur les centroïdes. Pour rendre l'évaluation plus correcte, nous recherchons actuellement des fonctions d'évaluation alternatives, soit directement définies sur les CE3D, soit personnalisées à partir de métriques génériques pour les contraintes sur les paires (Liu et al., 2007). Aussi nous cherchons à compléter notre évaluation sur une collection de documents techniques, qui est une tâche plus ambitieuse à cause de sa grande taille et de la forme complexe de ses clusters.

6 Conclusion

Nous avons décrit un système interactif dans lequel nous associons le clustering visuel dans un système 3D en coordonnées en étoiles avec le clustering semi-supervisé basé sur l'apprentissage d'une métrique de distance optimale à partir des retours de l'utilisateur. Le processus de clustering est guidé par l'utilisateur de façon intuitive et flexible ; ce dernier peut soit accomplir le clustering des données manuellement, soit le déléguer au système par l'annotation d'éléments ou l'indication de paires de similarité/dissimilarité. Différentes méthodes d'optimisation ont été comparées et une interface a été développée pour des tests en situation réelle. Il s'avère qu'associer le clustering visuel et les algorithmes automatiques offre un mécanisme puissant pour l'analyse intelligente des données complexes ou de grandes tailles.

Références

Basu, S., M. Bilenko, et R. J. Mooney (2004). A probabilistic framework for semi-supervised clustering. In *KDD'04 : Proc. 10th ACM SIGKDD Intern. Conf. Knowledge Discovery and Data Mining*, pp. 59–68.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Chen, K. et L. Liu (2004). Clustermap : labeling clusters in large datasets via visualization. In *CIKM'04 : Proc. 13th ACM Intern. Conf. Information and Knowledge Management*.

Chen, K. et L. Liu (2006). iVIBRATE : Interactive visualization-based framework for clustering large datasets. *ACM Trans. Inf. Syst. 24*(2), 245–294.

Chidlovskii, B., J. Fuselier, et L. Lecerf (2006). Aldai : active learning documents annotation interface. In *DocEng'06 : Proc. of the 2006 ACM Symp. on Doc. Engineering*.

Hastie, T., R. Tibshirani, et J. H. Friedman (2001). *The elements of statistical learning : data mining, inference, and prediction*. Springer.

Kandogan, E. (2001). Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *KDD'01 : Proc. 7th ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining*, pp. 107–116.

Liu, Y., R. Jin, et A. K. Jain (2007). Boostcluster : boosting clustering by pairwise constraints. In *KDD'07 : Proc. 13th ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining*, pp. 450–459.

Seo, J. et B. Shneiderman (2002). Interactively exploring hierarchical clustering results. *Computer 35*(7), 80–86.

Tang, W., H. Xiong, S. Zhong, et J. Wu (2007). Enhancing semi-supervised clustering : a feature projection perspective. In *KDD'07 : Proc. 13th ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining*, pp. 707–716.

Xing, E., A. Ng, M. Jordan, et S. Russell (2003). Distance metric learning, with application to clustering with side-information. *Advances in NIPS 15*.

Yu, L. et H. Liu (2004). Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res. 5*, 1205–1224.

Summary

In this paper we propose a method that combines the advanced data analysis of the automatic statistical methods and the flexibility and manual parameter tuning of interactive visual clustering. We present the *Semi-Supervised Visual Clustering* (SSVC) interface; its main contribution is the learning of the optimal projection distance metric for the *star and spherical coordinate* visualization systems. Beyond the conventional manual setting, it couples the visual clustering with the automatic setting where the projection distance metric is learned from the available set of user feedbacks in the form of either item similarities/dissimilarities or direct item annotations. Moreover, SSVC interface allows for the *hybrid* setting where some parameters are manually set by the user while the remaining parameters are determined by the optimal distance algorithm.